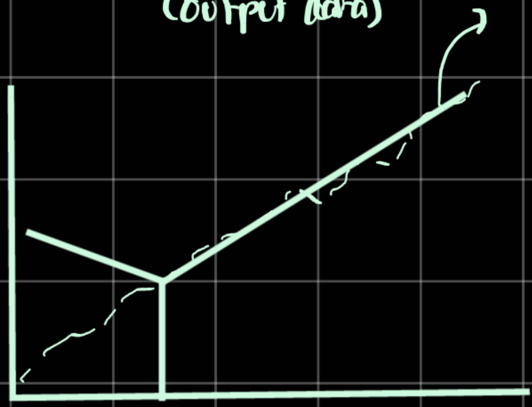
Deep learning
↓
Supervised → unsupervised → reinforced
↓
(AWS)

x (input data)

y (output data)

line that fits the dots say $F(x)$



$y = mx + c$

$F(x) = mx + c$

where m, c are parameters.

Observed values-

$x_1 \to y_1$

$x_2 \to y_2$

⋮

$x_n \to y_n$

To find $F(x)$ → the best possible line that describes $F(x)$

x	y	$F(x) \to y'$
x_1	y_1	$mx_1 + c$
x_2	y_2	$mx_2 + c$

Let $F(x)$ be y'

→ The error we get between the graph and ideal line is $|y-y'|$ → since abs is not differentiable

we take $\frac{d}{dx}(y-y')^2$ as the error.

∴ Error for each corresponding coordinate

$(x_1, y_1) \rightarrow \frac{d}{dx}(y - y_1')^2$

$(x_2, y_2) \rightarrow \frac{d}{dx}(y - y_2')^2$

⋮

for 'n' such elements

$$\sum_{i=1}^{n} \frac{d}{dx}(y - y_i')^2$$

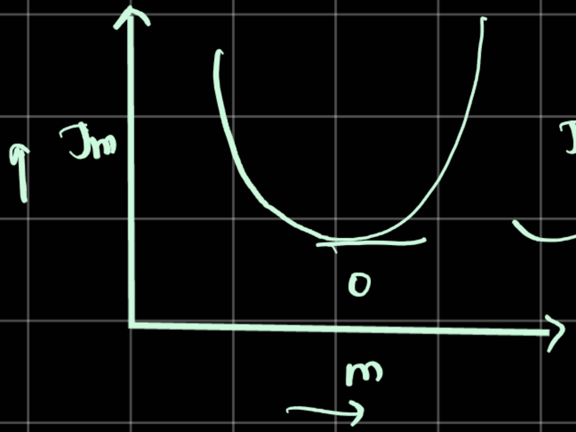
$$\sum_{i=1}^{n} \text{Error}$$

{ where $\frac{d}{dx}(y-y_i) =$ Error

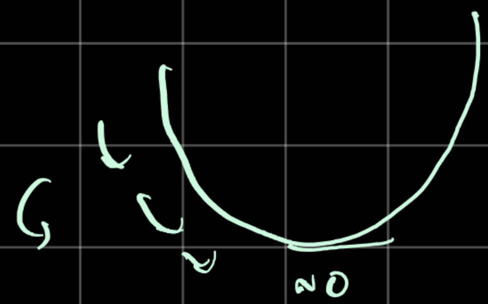
Thus, mean squared error is

$$\frac{1}{n} \sum_{i=1}^{n} \frac{d}{dx}(y-y_i')^2.$$

∴ Loss function $J_{m,c}() = \frac{1}{n} \sum_{i=1}^{n} \frac{d}{dx}(y - y_i')^2$



$J_{m,c} \simeq 0 \rightarrow$ The loss is minimum.



Here we are serching on the curve through descending thus, it's called gradient descent.

→ Loss function → multiple paramets

Say $F(x) = y' = mx_i + c$

$\downarrow$

$m, c \rightarrow$ parameters.

Thus, we do a partial derivative.

Err $J = \frac{1}{2n} \sum_{i=1}^{n} (y_i - (mx_i + c)^2)$

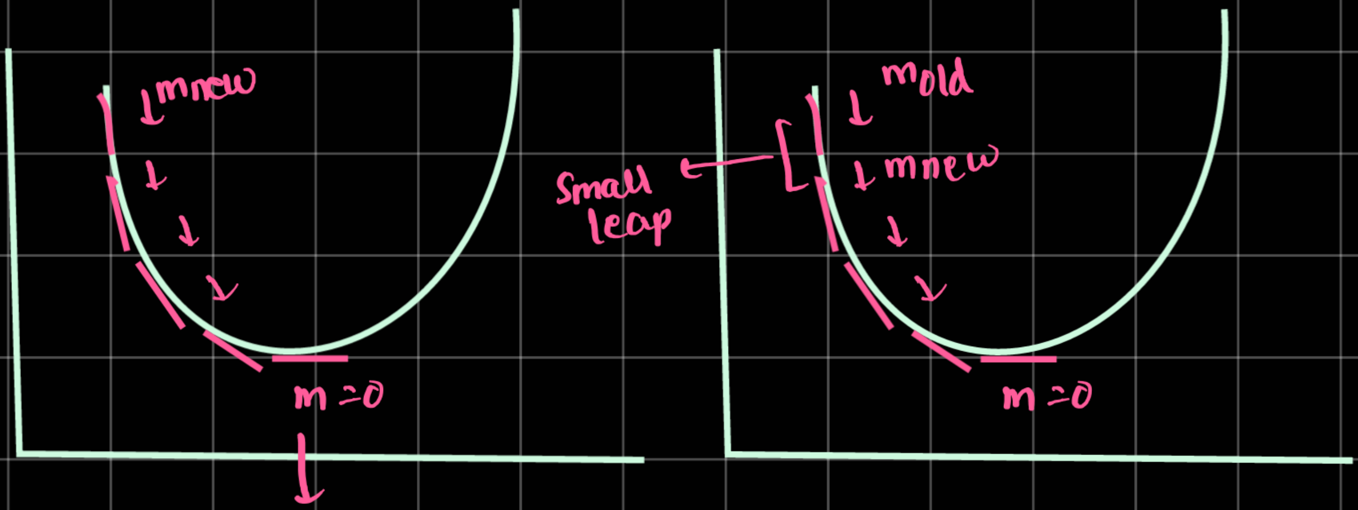
$\frac{\partial J}{\partial m} = ?$ $\frac{\partial J}{\partial c} = ?$

$$\frac{\partial J}{\partial m} = \frac{1}{\cancel{2}n} \sum_{i=1}^{n} \cancel{2}(y_i - (mx_i + c))(-x_i).$$

$$\frac{\partial J}{\partial m} = \frac{1}{\cancel{2}n} \sum_{i=1}^{n} \cancel{2}(y_i - (m x_i + c))(-1)$$

↳ $m_{new} = m_{old} - \boxed{\frac{\partial J}{\partial m}} \eta$ ← Gradient descent.

$c_{new} = c_{old} - \boxed{\frac{\partial J}{\partial c}} \eta$



→ Here we are traversing the curve till we reach $m=0$

→ For which we are optimising the direction → through sign of slope

length of traversal → by finding optimum constant through partial diff

↳ Since it is difficult to calculate every possible

Gradient descent

→ Stochastic G.D

→ ** Batch G.D → Random Batch

↓

Each batch is representative of entire data

↓

4, 16, 32, 64, 512, 1024.

→ $x_1 \quad x_2 \quad x_3 \dots x_n$

$\theta_0, \theta_1 x_1, \theta_2 x_2 \dots \theta_n x_n$

↳ feature Engineering

Loss function → how far mathematically is my function from my ideal value.